

Association for Information Systems AIS Electronic Library (AISeL)

Wirtschaftsinformatik Proceedings 2005

Wirtschaftsinformatik

February 2005

Benutzerorientierter Entwurf von unternehmensweiten Data-Warehouse-Systemen

Lars Burmester

Philipps-Universität Marburg

Matthias Goeken

Philipps-Universität Marburg

Follow this and additional works at: <http://aisel.aisnet.org/wi2005>

Recommended Citation

Burmester, Lars and Goeken, Matthias, "Benutzerorientierter Entwurf von unternehmensweiten Data-Warehouse-Systemen" (2005).
Wirtschaftsinformatik Proceedings 2005. 75.
<http://aisel.aisnet.org/wi2005/75>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2005 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

In: Ferstl, Otto K, u.a. (Hg) 2005. *Wirtschaftsinformatik 2005: eEconomy, eGovernment, eSociety*;
7. Internationale Tagung Wirtschaftsinformatik 2005. Heidelberg: Physica-Verlag

ISBN: 3-7908-1574-8

© Physica-Verlag Heidelberg 2005

Benutzerorientierter Entwurf von unternehmensweiten Data-Warehouse-Systemen

Lars Burmester, Matthias Goeken

Philipps-Universität Marburg

Zusammenfassung: Der Beitrag beschreibt eine Entwurfsmethode für die Data-Warehouse-Entwicklung. Als Bezugsrahmen dient dabei eine erweiterte Data-Warehouse-Architektur und ein Verfahren zur Zerlegung des zu erstellenden Systems. Dies unterstützt die Definition von Ausbaustufen für ein inkrementelles Vorgehen. Für die einzelnen Inkremente werden ausgehend vom Informationsbedarf der Benutzer fachkonzeptionelle Strukturen entwickelt, die wiederum schrittweise in logische Schemata transformiert werden. Diese logischen Modelle dienen der Vorbereitung der Implementierung auf physischer Ebene und der Modellierung der ETL-Prozesse.

Schlüsselworte: Data-Warehouse-System, Multidimensionale Modellierung, ETL-Modellierung, Inkrementelles Vorgehen, Informationsbedarfsanalyse

1 Einleitung

Die Entwicklung von Data-Warehouse-Systemen steht regelmäßig im Spannungsverhältnis zwischen betriebswirtschaftlich-fachlichen und technisch-integrativen Anforderungen. Die zentrale Herausforderung besteht daher darin, eine informationsbedarfs- und benutzerorientierte Sichtweise mit einer am Informationsangebot ausgerichteten technischen Sichtweise abzustimmen. Grundsätzlich bieten sich hierfür zwei generelle Vorgehensrichtungen an. Bei einer am Informationsbedarf ausgerichteten Vorgehensweise würde die Entwicklung des Gesamtsystems auf Ebene der inhaltlichen Anforderungen an das System beginnen. Die Entwicklung der Data-Warehouse-Ebene oder der ETL-Prozesse würde nachlaufend erfolgen, sodass diese Art des Vorgehens auch als Top-Down-Vorgehen bezeichnet werden kann. Im Gegensatz dazu stellen bei einer Bottom-Up-Vorgehensweise z.B. die konzeptionellen Schemata der operativen Systeme den Ausgangspunkt dar, aus denen die Schemata des Data-Warehouse abgeleitet werden [Gol⁹⁸; GoRi98].

Legt man ein Top-Down-Vorgehen zu Grunde, so stellt die Analyse des Informationsbedarfs regelmäßig das Kernproblem dar [Holt99]. Als Hindernisse bei der Erhebung des Informationsbedarfs identifizieren Valusek und Fryback „obstacles

which can be categorized as *within* an individual user, *among* users, and *between* the individual user and those responsible for system development“ [VaFr85, ähnlich BrRo01]. Die Between-Obstacles ergeben sich dadurch, dass Entwickler und Benutzer in unterschiedlichen Sprachen sprechen. Hieraus resultiert eine „Sprachlücke“, die durch geeignete Methoden und Techniken überwunden werden muss [Ortn95]. Zusätzlich sind die Anforderungen und Informationsbedarfe im Verlaufe eines Entwicklungsprojekts häufig noch instabil, da sie der Dynamik des betrieblichen Umfelds unterliegen und die zukünftigen Benutzer eines Systems sich möglicherweise erst im Projektverlauf ihrer Bedarfe klar werden.

Den skizzierten Problemen soll im Folgenden durch ein prototypenorientiertes Vorgehen begegnet werden. Dabei werden lauffähige Prototypen eingesetzt, um bestimmte Aspekte des zu realisierenden Systems zu veranschaulichen und so auf einfache, realitätsnahe Weise mit dem zukünftigen Anwender zu kommunizieren [Hess97; Hec⁺03; Bey⁺99].

Die zentrale technische Herausforderung im Data Warehousing besteht darin, einen unternehmensweit integrierten und konsistenten Datenbestand aufzubauen, was regelmäßig ein komplexes, langwieriges und kostspieliges Unterfangen mit einem hohen Realisationsaufwand darstellt [Eick01; Hack98]. Daher wird häufig empfohlen, mit der Entwicklung von Data Marts zu beginnen, also inkrementell vorzugehen und zunächst kleine Data-Warehouse-Systeme für einzelne Funktionsbereiche oder Abteilungen zu erstellen. Unterstützung für eine ganzheitliche Planung unternehmensweiter Data-Warehouse-Systeme findet sich jedoch nur vereinzelt (vgl. zu älteren Ansätzen der Architekturevolution [Hack98; Ong99]).

In den folgenden Ausführungen wird eine auf den einleitenden Überlegungen basierende Entwurfsmethode für Data-Warehouse-Systeme vorgestellt. In Kapitel 2 wird eine erweiterte Data-Warehouse-Architektur hergeleitet, welche als Bezugsrahmen für den Entwurfsprozess dient. Kapitel 3 stellt die Entwurfsmethode dar, wobei die Planung des Gesamtsystems und das Vorgehen einleitend erläutert werden (Kapitel 3.1). Die Analyse und der Entwurf der fachlichen Strukturen sowie deren Validierung vor dem Hintergrund der Benutzeranforderungen wird in Kapitel 3.2 dargestellt. Dieser dient der Überwindung der oben erwähnten Sprachlücke in der Benutzer-Entwickler-Kommunikation. In Kapitel 3.3 erfolgt eine Beschreibung des Übergangs zur Systemspezifikation und der Implementierung der ETL-Prozesse. Abschließend soll ein kurzer Ausblick erfolgen (Kapitel 4).

2 Erweiterte Data-Warehouse-Architektur

In diesem Abschnitt wird eine erweiterte Data-Warehouse-Architektur vorgestellt, die die Entwicklungsergebnisse der in Abschnitt 3 beschriebenen Methode präsentiert. Durch die Betrachtung der Architekturebenen eines Data-Warehouse-

Systems lässt sich der schrittweise Datenfluss von den Quellsystemen hin zu analyseorientierten Anwendungen auf der Präsentationsebene beschreiben (vgl. dazu Abbildung 1, rechte Seite sowie [Holt99; Herd01; Sche00; BoU100]). Mit dieser physischen Perspektive wird jedoch nur ein Teil der Aufgaben des Data Warehousing abgedeckt. Insbesondere lassen sich die fachlich-betriebswirtschaftlichen Aufgaben eines Data-Warehouse-Systems in dieser nicht adäquat fassen [Vas⁺00; Jar⁺99; Quix03].

Um die Datenbestände der operativen Systeme nicht nur physisch, sondern auch in fachlicher Hinsicht für die Informationsversorgung und Entscheidungsunterstützung nutzbar zu machen, bietet sich an, die jeweiligen Architekturebenen in eigenen konzeptionellen Schemata explizit zu beschreiben. Dadurch wird neben dem (Bottom-up-) Datenfluss der Prozess der Nutzung von Informationen mit in die Betrachtung integriert. Die konzeptionelle Perspektive bzw. die konzeptionellen Schemata unterstützen die Kommunikation mit Benutzern / Anwendern und dokumentieren Ergebnisse der einzelnen Entwicklungsschritte. Sie sind benutzer- und anwendernah, da sie Modellierungskonstrukte verwenden, die für diese leicht verständlich sind. Zusätzlich stellen sie den entscheidenden „Input“ für die folgenden Phasen des Entwicklungsprozesses dar [WaWe02].

Den Ausgangspunkt bilden konzeptionelle multidimensionale Datenschemata (MDDS), die der Erhebung der informatorischen Anforderungen an das zu erstellende System dienen (Abbildung 1). Sie unterstützen die Informationsbedarfsanalyse und dokumentieren deren Entwicklungsergebnisse. Hierbei wird zwischen den Anforderungen eines Benutzers (bzw. einer Benutzergruppe) und den Anforderungen, die schließlich in die Spezifikation des Gesamtsystems eingehen, unterschieden. Diese Unterscheidung korrespondiert mit der für FIS geforderten Empfängerorientierung und erlaubt die Rückverfolgung (Traceability) von Anforderungen hin zu ihrer Quelle (vgl. [Goek04a]; dort auch zu nicht-informatorischen Anforderungen, wie z.B. Performance, Usability usw.). Durch die individuellen konzeptionellen MDDS werden zum einen adressatengerechte Berichte, Navigationsmöglichkeiten und Alternativen zur Analyse des Entscheidungsraumes definiert, die durch die Frontendclients bereitgestellt werden. Zum anderen werden sie konsolidiert und zum konzeptionellen MDDS vereint, welches wiederum in das logische MDDS transformiert wird. Darüber hinaus werden konzeptionelle Schemata auch für die Datenerfassungsebene entworfen. Diese konzeptionellen Schemata der ETL-Prozesse, die aus den logischen MDDS abgeleitet werden, dienen der Kommunikation mit den Administratoren der operativen Systeme. Sie stellen gleichsam Anforderungen des Data-Warehouse-Systems an diese als Datenlieferanten dar. Die konzeptionellen Schemata werden wiederum in logische Schemata transformiert und physisch als ETL-Prozesse implementiert.

Durch diese erweiterte Architektur ergibt sich ein durchgängig strukturierter Prozess zur Entwicklung von Data-Warehouse-Systemen und deren Dokumentation auf verschiedenen semantischen Stufen. Darüber hinaus stellt die Architektur ei-

nen Bezugsrahmen dar, der eine Integration verschiedener, z. T. unabhängiger Modellierungsansätze im Gebiet des Data Warehousing ermöglicht.

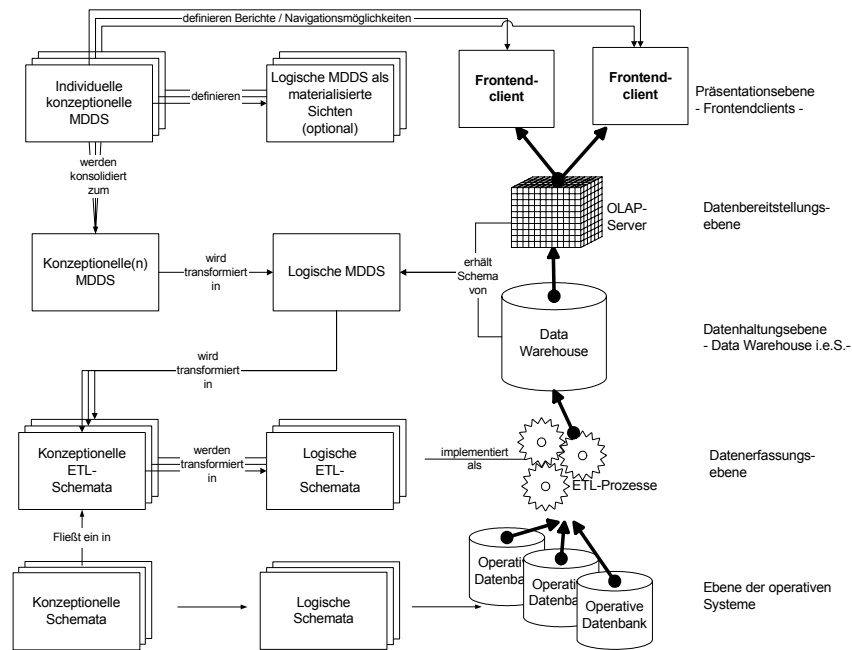


Abbildung 1: Erweiterte Data-Warehouse-Architektur¹

Die so beschriebene erweiterte Data-Warehouse-Architektur weist Parallelen zu dem im DWQ-Projekt entwickelten „Meta-Data-Framework“ auf [Vas⁺00; Jar⁺99; Quix03]. Sie unterscheidet sich von diesem jedoch in zwei wesentlichen Punkten: Zum einen wird im DWQ-Projekt ein „Enterprise Model“ als gegeben angenommen. Da solche Unternehmensdatenmodelle in der Praxis oftmals nicht vorhanden sind und in der Wirtschaftsinformatik häufig als problematisch angesehen werden [StHa05; Schi01], wird ein solches an dieser Stelle nicht als gegeben angenommen und auch nicht hergeleitet. Den Ausgangspunkt bilden stattdessen die individuellen Anforderungen der Benutzer und Anwender. Zum zweiten wird ein anderer Integrationsansatz verfolgt. Während im DWQ-Projekt der sog. Local-as-View-Ansatz zugrunde gelegt wird, werden hier die Inhalte des Data-Warehouse als Sichten auf die Datenquellen beschrieben. Dieser Global-as-View-Ansatz ent-

¹ Zur Vereinfachung wird davon ausgegangen, dass sich die Schemata der Datenbereitstellungsebene und der -haltungsebene entsprechen. D.h. dass das Data Warehouse i. e. S. bereits die für den OLAP-Server notwendigen Datenstrukturen enthält. Soll dagegen auf der Datenhaltungsebene ein „Reconciled Data Layer“ eingerichtet werden, dann greifen die in Teil 3.3 beschriebenen Transformations- und Ladeprozesse auf dieses zu anstatt auf die operativen Systeme.

spricht einem prozeduralen Vorgehen, mit dem sich besser beschreiben lässt, wie die Daten für das integrierte System aus den Datenquellen extrahiert und geladen werden.

3 Methode zum benutzerorientierten Entwurf von Data-Warehouse-Systemen

3.1 Ausbaustufenplanung und Vorgehen

Um die Komplexität der Entwicklung unternehmensweiter Data-Warehouse-Systeme zu verringern, wird oftmals ein von Data-Marts ausgehendes, inkrementelles Vorgehen vorgeschlagen, welches eine Orientierung an einem festen Ziel bedeutet, das durch die schrittweise Erweiterung von zunächst unvollständigen Teillösungen verfolgt wird [Hess97; Flo⁺97]. Dies erlaubt einen flexiblen Umgang mit der Dynamik des Umfeldes und ermöglicht eine feingranulare Terminplanung. Missverständnisse können frühzeitig beseitigt werden, und die Erprobung liefert Hinweise, ob das zur Verfügung gestellte Informationsangebot dem tatsächlichen Informationsbedarf entspricht. Eine inkrementelle Vorgehensweise macht es erforderlich, dass das zu entwickelnde Gesamtsystem zerlegbar ist. Ist dies gegeben, können Ausbaustufen (Builds) eines Systems festgelegt und in einem Ausbaustufenplan definiert werden [Flo⁺97; GoLi93]. Die sich so ergebenden Schleifen im Entwicklungsprozess können als geplante Iterationen angesehen werden.

Bei dem hier vorgestellten Vorgehen erfolgt eine Zerlegung des zu erstellenden Data-Warehouse-Systems anhand von Informations- und Entscheidungsobjekten auf der einen Seite und Architekturebenen auf der anderen Seite. Die Informations- und Entscheidungsobjekte stellen betriebswirtschaftlich relevante Betrachtungsgegenstände dar, die bspw. aus dem Zielsystem des Unternehmens oder einem Controllingkonzept abgeleitet werden können. Sie definieren die thematischen Schwerpunkte, an denen sich das zu erstellende Data-Warehouse-System orientiert, auf abstrakter Ebene. Durch Kombination dieser fachlichen Sicht mit den Architekturebenen eines Data-Warehouse-Systems ergeben sich Entwicklungsobjekte, die Module des Gesamtsystems darstellen. (vgl. Abbildung 2).

Finanzen	Vertrieb	Produktion	...	Personal	
					Präsentationsebene
					Datenbereitstellungsebene
					Datenhaltungsebene
					Datenerfassungsebene

Abbildung 2: Entwicklungsobjekte als Ergebnis der Systemzerlegung

Im Rahmen der Prototypenplanung sowie der Planung von Ausbaustufen bei der inkrementellen Entwicklung stellen diese Entwicklungsobjekte den Gegenstand von Entscheidungen dar. D. h. es wird festgelegt, welche Objekte in welcher Reihenfolge realisiert werden sollen. Um eine bessere Feinplanung der Entwicklungsstufen eines Entwicklungsobjekts zu ermöglichen, erfolgt dessen Betrachtung in Form der vorgestellten erweiterten Data-Warehouse-Architektur. Im Ausbaustufenplan können Zeitpunkte (Milestones) im Entwicklungsprozess definiert werden, zu denen ein bestimmter Entwicklungsvorgang abgeschlossen sein muss. Abbildung 3 stellt den Ausbaustufenplan eines beispielhaften Projekts dar.

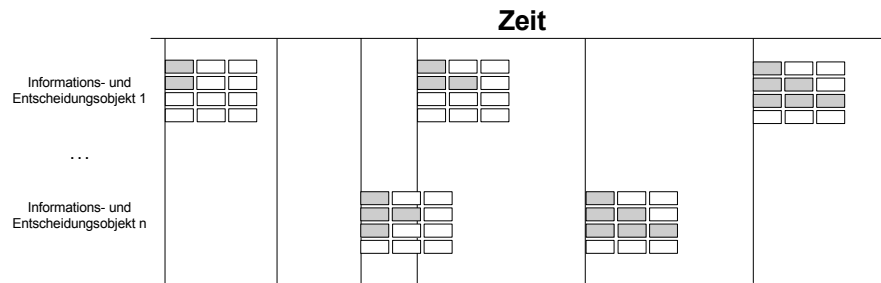


Abbildung 3: Beispielhafter Ausbaustufenplan verschiedener Teilsysteme

Das Vorgehen zur Entwicklung einzelner Ausbaustufen gliedert sich in mehrere Teilschritte. Den Ausgangspunkt stellt die Wissensakquisitionsphase dar, in welcher die inhaltlichen Anforderungen der einzelnen Nutzer an das zu entwickelnde System erhoben werden. Zu Beginn der Entwurfsphase werden diese Anforderungen in individuelle konzeptionelle MDDS überführt und im weiteren Phasenverlauf zu einer konzeptionellen Gesamtsicht konsolidiert (konzeptionelles MDDS). Den Abschluss der Entwurfsphase bildet die Transformation des konzeptionellen MDDS in ein logisches Datenmodell (logische MDDS), welches als Grundlage für die Konstruktion eines Validierungsprototypen dient. In der Validierungsphase erfolgt die Validierung der bislang generierten fachlichen Strukturen durch die zukünftigen Nutzer, wobei dies anhand des in der vorangegangenen Phase konstruierten Prototyps geschieht. Bei Ablehnung des Prototyps sind in Abhängigkeit von den geäußerten Ablehnungsgründen explizite Rückschritte bzw. Rückkopplungen in frühere Phasen vorgesehen. Die positive Validierung des Prototyps ist Auslöser für den Übergang zur Spezifikation und Implementierung des Systems. Hierbei spezifizieren die generierten logischen MDDS die formalisierte Informationsnachfrage, welche mit dem Informationsangebot auf der Datenerfassungsebene zum Ausgleich gebracht werden soll. Die Transformation der logischen MDDS in konzeptionelle ETL-Schemata sowie die darauf basierende logische Abbildung von ETL-Prozessen bilden die Grundlage für die abschließende physische Implementierung. Die Gesamtsicht auf das Vorgehensmodell für die Realisierung der einzelnen Entwicklungsobjekte stellt Abbildung 4 dar.

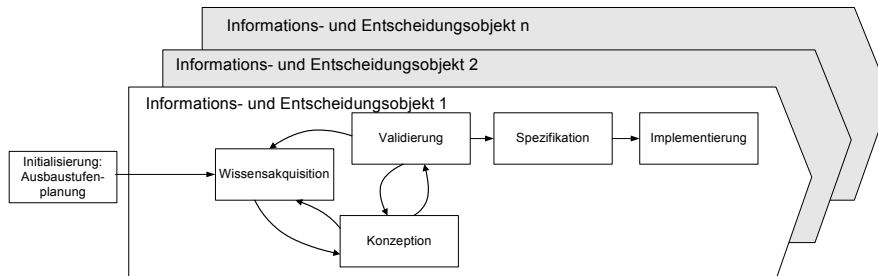


Abbildung 4: Vorgehensmodell für die Realisierung einzelner Entwicklungsobjekte

3.2 Analyse und Entwurf der fachlichen Strukturen

3.2.1 Wissensakquisition

Die Phase Wissensakquisition stellt den Einstieg in den Zyklus zur Ermittlung und Formulierung des bestehenden Informationsbedarfs dar. Dieser wird zunächst einzeln für die zukünftigen Nutzer oder ggf. für Nutzergruppen erhoben. Hierbei soll einerseits festgestellt werden, welche Fakten und Kennzahlen ein Informations- und Entscheidungsobjekt quantifizieren. Andererseits werden mit der Ermittlung von Dimensionen und Dimensionshierarchien benutzerdefinierte, qualifizierende Sichten auf die Maßgrößen bzw. Kennzahlen geschaffen [Lehn03].

Aufgrund des hohen Abstraktionsgrades erscheinen die gängigen konzeptionellen multidimensionalen Modellierungssprachen (z.B. DFM, ADAPT u. a.; vgl. [Bulo98; Gol⁺98; Abe⁺00]) zum Teil als ungeeignet, um mit dem Nutzer über die Anforderungen an das zu entwickelnde System zu kommunizieren. Die mit konzeptionellen Modellen verfolgten Ziele – Dokumentation, Input für den weiteren Entwicklungsprozess und Kommunikation mit dem Benutzer [WaWe02] – sind nach Ansicht der Autoren nur schwer mit ein und derselben Sprache zu realisieren. Vor allem bei der Überwindung der „Sprachlücke“ ergeben sich Probleme, da die formale Darstellung häufig die Artikulation des Informationsbedarfs erschwert. Insbesondere erweist es sich oft als schwierig, den Benutzern die Unterschiede der in der multidimensionalen Datenmodellierung verwendeten Konstrukte nahe zu bringen. (Zu einer empirischen Studie, die zu ähnlichen Ergebnissen gelangt vgl. [NoCr99]). Somit bleibt es Aufgabe des Entwicklers, die geäußerten Bedarfe der Benutzer zu analysieren und zu interpretieren. Hierzu und auch zur Kommunikation haben sich sog. „Interrogatives“ bzw. „W-Fragen“ (was, wann, wer, wo, womit ...) als nutzbringend erwiesen, da sie gebrauchssprachlich die Konstrukte der multidimensionalen Modellierung beschreiben [vgl. BrRo01; StHa05; QuDe99]. Interrogative lassen sich als eine einfache Grammatik (Satzbauplan) interpretieren, wie sie von Ortner für den methodenneutralen Fachentwurf von Anwendungssystemen vorgeschlagen wird [Ortn95].

Zur Repräsentation wird ein vereinfachtes konzeptionelles Modell (dimensionales Kennzahlenmodell) genutzt, das nur eine Teilmenge der Modellelemente verwendet, die in konzeptionellen multidimensionalen Modellen zum Einsatz kommen (Fakten, Kennzahlen und Dimensionen (vgl. Abbildung 5)). Diese Beschränkung auf wenige Konstrukte führt zu einem besseren Verständnis und ermöglicht die Konzentration auf allgemeine Aspekte. Im oberen Teil des vereinfachten konzeptionellen Modells findet sich ein Kennzahlensystem, welches die relevanten Fakten des Entwicklungsobjekts sowie die eigentlichen Kennzahlen beinhaltet. Zusätzlich werden, da in betriebswirtschaftlichen Ansätzen die Dimensionen von Kennzahlen in der Regel keine Beachtung finden, diese im unteren Teil ergänzt. Hierbei ist zu beachten, dass versucht wird, Dimensionen gleicher Art auf einer Ebene anzuordnen. Dadurch werden Überschneidungen der Dimensionen verschiedener Kennzahlen deutlich. Für das konzeptionelle Schema ergeben sich so potenziell gemeinsame Dimensionen, bzw. Conformed Dimensions [KiRo02].

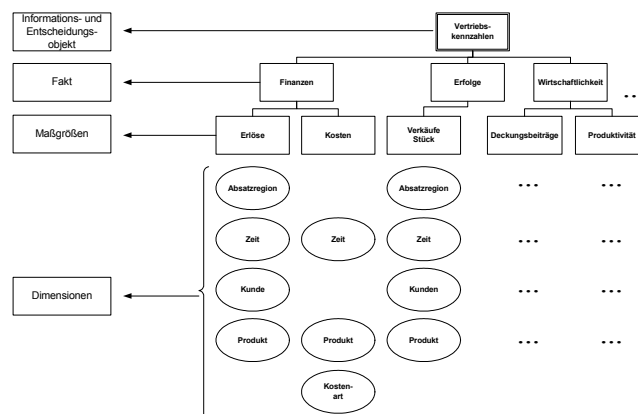


Abbildung 5: Vereinfachtes konzeptionelles Modell (dimensionales Kennzahlenmodell)

Das Ergebnis der Wissensakquisitionsphase stellt die Repräsentation der Informationsbedarfe einzelner Benutzer und Benutzergruppen dar. Als Artefakte bzw. Ergebnisdokumente gehen eine Reihe individueller vereinfachter konzeptioneller Schemata sowie „Interrogatives“ in die nächste Phase ein.

3.2.2 Konzeption

Im Folgenden soll der Entwurfsprozess im Rahmen der erweiterten Data-Warehouse-Architektur vorgestellt werden, welcher die informatorischen Anforderungen, die im Rahmen der Wissensakquisitionsphase ermittelt wurden, umsetzt. Den Ausgangspunkt bilden hierbei die Ergebnisdokumente der Wissensakquisitionsphase, welche die Bedarfe einzelner Anwender widerspiegeln. In einem ersten Schritt erfolgt die Transformation der Ergebnisdokumente in gängige konzeptionelle MDDS sowie deren Konsolidierung zu einer Gesamtsicht auf das Sys-

tem. Da diese Modelle nicht mehr zur direkten Kommunikation mit dem Nutzer herangezogen werden, entfallen auch die bisher vorgebrachten Einwände. Vielmehr dienen sie neben der Dokumentation als Input für die weitere Transformation in ein logisches MDDS (vgl. wiederum Abbildung 1).

Zur Illustration des Vorgehens soll ein durchgängiges Beispiel herangezogen werden. Dabei handelt es sich um Elemente eines Vertriebsinformationssystems, wobei der Bereich Finanzen im Mittelpunkt steht. Im Rahmen der Wissensakquisitionsphase wurden die Einzelsichten der Benutzer auf das zukünftige System erhoben und in vereinfachten konzeptionellen Modellen zusammengefasst. Des Weiteren kann das zu entwickelnde System mit dem Interrogativ „Welche Beträge (in €), welcher Art (z.B. Erlöse, Erlösschmälerungen, Promotion-Kosten), sind bei welchen Kunden (Großhandel, Einzelhandel usw.), in welcher Region (Bundesland, Nielsen-Gebiet, usw.), wann (Kalenderjahr, Fiskaljahr) geflossen?“ charakterisiert werden. Als Datenquellen sollen Verkaufsdaten aus dem ERP-System sowie Daten aus nicht integrierten Marketinginformationssystemen (z.B. Systeme zur Werbe- und Promotionplanung) herangezogen werden.

Die Abbildung der dimensional Kennzahlenmodelle in eine gängige Modellierungssprache soll an dieser Stelle beispielhaft an der Transformation in die Notation des ADAPT dargestellt werden [BuFo02]. Den Ausgangspunkt bilden hierbei die Dimensionen, die ein Benutzer im Rahmen der Wissensakquisition für die Beschreibung einer Kennzahl genannt hat. Die Formalisierung einer Dimension erfordert die Darstellung der hierarchischen Strukturen der Dimensionselemente. In gängigen multidimensionalen Modellierungssprachen werden Dimensionshierarchien nicht vollständig abgebildet, da dies das Modell schnell unübersichtlich werden lässt. Daher werden Dimensionen meist komprimiert, in Form generalisierter, abstrahierender Dimensionsebenen dargestellt. Einfache, balancierte Hierarchien können direkt in diese Darstellungsform überführt werden (vgl. Abbildung 6). Bei Dimensionen mit komplexeren Dimensionsstrukturen, wie z.B. multiplen oder unbalancierten Hierarchien sowie anteiligen Verrechnungen, kann dieses Verfahren nicht ohne Weiteres angewandt werden, sodass ggf. wieder von einer stark komprimierten Abbildung abgewichen werden muss [Sche98].

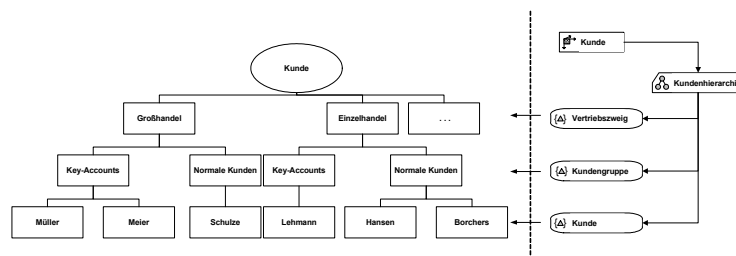


Abbildung 6: Transformation einer einfachen balancierten Dimensionshierarchie in die Notation des ADAPT.

Nach der Überführung der dimensional Modelle der Einzelsichten in formalisierte konzeptionelle MDDS sollen diese zu einer Mehrpersonensicht konsolidiert werden. Das daraus resultierende Schema stellt das konzeptionelle MDDS der Datenhaltungsebene dar. Da die Maßgrößen und Kennzahlen des Faktis in der Regel bekannten betrieblichen Kennzahlensystemen entstammen, sollte ihre Konsolidierung ohne größeren Aufwand zu realisieren sein. Dahingegen stellt die Zusammenführung der Dimensionen eine zentrale Herausforderung dar. Hierbei ist es erforderlich, unterschiedliche Sichten auf Dimensionen zu erkennen und zu einer konsistenten Dimension zu konsolidieren. Im Folgenden soll dieser Prozess an unterschiedlichen Sichten auf eine Zeitdimension demonstriert werden. In diesem fiktiven Beispiel benötigt ein Controller für seine Analysen deutlich tiefere Untergliederungen der Zeitdimension als bspw. der Vertriebsvorstand des Unternehmens. Letzterer orientiert sich am Fiskaljahr, während der Controller das Kalenderjahr für seine Analysen zu Grunde legt. Abbildung 7 zeigt die im Beispiel skizzierte Konsolidierung verschiedener Sichten auf eine Zeitdimension.

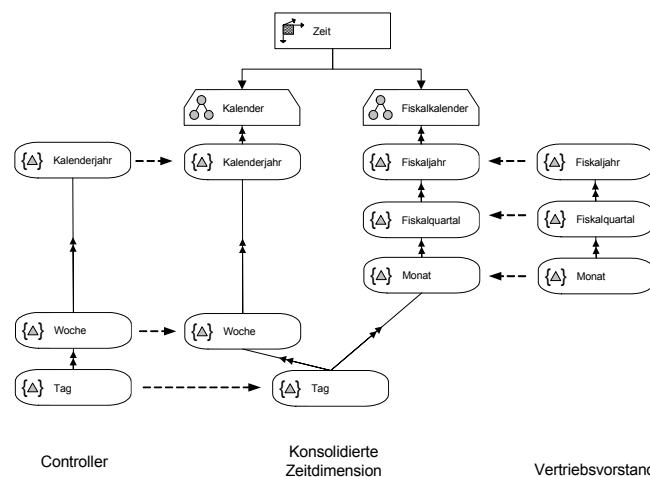


Abbildung 7: Konsolidierung verschiedener Sichten auf eine Zeitdimension

Nach Bildung des konzeptionellen MDDS der Mehrpersonensicht erfolgt dessen Transformation in das logische Schema der Datenhaltungsebene. Trotz der Unabhängigkeit von der späteren physischen Repräsentation sind logische Modelle multidimensionaler Strukturen an der einzusetzenden Datenbanktechnologie ausgerichtet. Aufgrund des hohen Verbreitungsgrads relationaler Datenbanken sind auf Ebene der logischen Modellierung regelmäßig Varianten des Star-Schemas anzutreffen [Hahn98], weshalb diese im Folgenden im Mittelpunkt stehen.

Die Bildung eines logischen Datenschemas aus konzeptionellen Strukturen stellt einen kritischen Schritt im Entwicklungsprozess dar, da diese Transformation mit einem semantischen Informationsverlust verbunden ist [Sche00; Blas00; Hahn02]. Daher ist an dieser Stelle auch die Wahrscheinlichkeit von Mappingfehlern am

größten. Betrachtet man die grundlegenden Bestandteile eines multidimensionalen Datenmodells, erscheint die Umwandlung von Fakten bzw. Maßgrößen im Vergleich zur Transformation der Dimensionen in wenigen Schritten realisierbar. Obwohl die meisten semantischen Modellierungssprachen in der Lage sind, komplexe Dimensionstrukturen, wie z.B. multiple oder unbalancierte Hierarchien, anteilige Verrechnungen oder nicht additive Roll-ups [Herd01], abzubilden, existieren kaum Ansätze, die deren semantisch verlustfreie Übertragung in das logische Modell zulassen.

Das Mapping eines konzeptionellen multidimensionalen Modells auf logische ROLAP-Schemata soll im Folgenden durch Betrachtung ihrer Metamodelle erläutert werden [Blas00]. Nahezu alle multidimensionalen Modellierungssprachen verfügen über grundlegende Konstrukte wie Fakt, Maßgrößen, Dimensionsebenen und Dimensionsattribute sowie über eine Notation zur Abbildung der hierarchischen Beziehungen innerhalb einer Dimension. Auch die Beziehungen dieser Konstrukte untereinander sind festgelegt, z.B. dass einem Fakt eine oder mehrere Dimensionen zugeordnet sind. Diese Konstrukte und Beziehungen lassen sich in Form eines ERM darstellen, wobei Kardinalitäten zur Beschreibung der Beziehungen eingesetzt werden. Da das Ziel der Transformation eine Relation ist, soll diese ebenfalls in ihren grundlegenden Konstrukten und Beziehungen beschrieben werden. Als Entity-Typen gelten hierbei die Tabelle selbst sowie die Spalte. Diese werden durch eine 1:n-Beziehung miteinander verbunden.

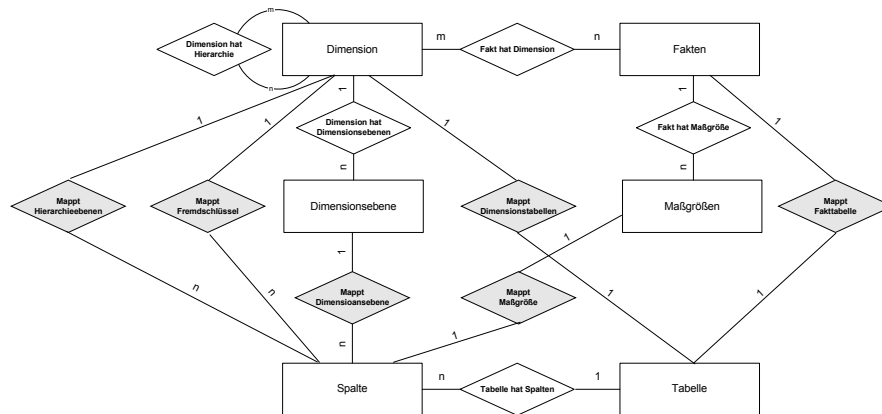


Abbildung 8: Metamodel der Mappingbeziehungen zwischen konzeptionellem und logischem Modell (verändert übernommen aus [Blas00, S. 92])

Das Mapping des multidimensionalen Modells auf Tabellenstrukturen lässt sich in Form von Beziehungen zwischen den einzelnen Metamodellen darstellen. So stehen beispielsweise die Entity-Typen Fakt oder Dimension in einer 1:1-Beziehung mit dem Entity-Typ Tabelle (für jede Dimension bzw. für jedes Fakt existiert genau eine Tabelle). Dagegen stehen Entity-Typen wie Dimensionsebene und Maßgröße in Beziehung mit dem Entity-Typ Spalte der korrespondierenden Tabelle

(Für jede Maßgröße existiert genau eine Spalte in der Fakttable). Eine Darstellung der Metamodelle (weiß) sowie der Mappingbeziehungen zwischen ihnen (grau hinterlegt) kann Abbildung 8 entnommen werden.

Aus den dargestellten Mappingbeziehungen können Regeln zur Transformation eines konzeptionellen Modells hergeleitet werden [vgl. GoRi98; Herd01]. So kann die 1:1-Beziehung zwischen Fakt und Dimension als „Bilde für jeden Fakt und jede Dimension genau eine Tabelle“ formuliert werden. Dies führt bei einem einfachen multidimensionalen Schema zu einer Fakt- und mehreren Dimensionstabellen (den grundlegenden Bestandteilen eines Star-Schemas). Die Spaltenbildung in einer Tabelle erfolgt in Abhängigkeit vom Tabellentyp. Bei Dimensionstabellen werden Spalten für Dimensionsebenen, hierarchische Beziehungen sowie für einen Primärschlüssel gebildet. Für die Fakttable müssen Spalten für die Maßgrößen sowie die Primärschlüssel der Dimensionstabellen als Fremdschlüssel eingesetzt werden. Die Verknüpfung zwischen Fakttable und Dimensionstabellen erfolgt über die vergebenen Fremdschlüssel, wobei die Menge der Fremdschlüssel den zusammengesetzten Primärschlüssel der Fakttable ergeben. Abbildung 9 zeigt beispielhaft das Mapping eines ADAPT-Modells auf ein Star-Schema (logisches MDDS).

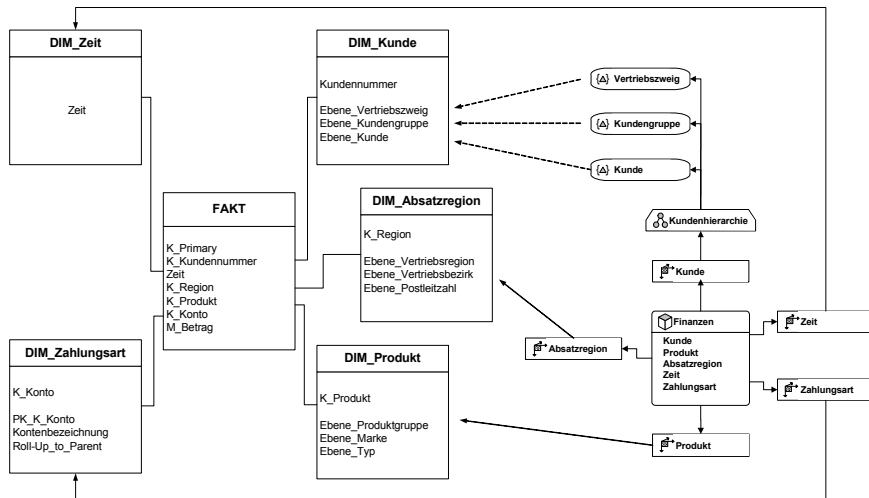


Abbildung 9: Mapping von ADAPT auf ein einfaches Star-Schema

Zur relationalen Optimierung einer ROLAP-Lösung hinsichtlich Performance, Wartbarkeit und Speicherplatz bieten sich Verfahren wie z.B. Indexierung, Partitionierung sowie die Materialisierung von Sichten an [Herd01; Pera03; PeRu03; BoU100; Lehn03]. Auf eine Darstellung und Diskussion der genannten Verfahren und ihrer Auswirkungen auf das Gesamtsystem soll in diesem Rahmen verzichtet werden, da an dieser Stelle lediglich die Umsetzung der Anforderungen, die sich aus dem konzeptionellen Modell ergeben, erläutert werden soll.

Durch das logische MDDS sind die inhaltlichen Anforderungen an ein Data-Warehouse-System in Tabellenstrukturen abgebildet. Zur Validierung der generierten Strukturen ist es erforderlich, diese mit Daten zu befüllen. Im Rahmen des hier vorgeschlagenen inkrementellen und prototypingorientierten Vorgehens sollte zunächst auf Simulationsdaten zurückgegriffen werden, da die Modellierung und Implementierung von ETL-Prozessen mit erheblichem Aufwand verbunden ist ([Vas⁺02a] bspw. schätzen den Anteil an der Gesamtentwicklung auf 80%).

3.2.3 Validierung

Die Validierung des konzeptionellen Modells und der logischen Strukturen erfolgt in einem gemeinsamen Schritt. Die Benutzer des zu erstellenden Systems können anhand des fertig gestellten Prototyps validieren, ob die erhobenen Anforderungen bezüglich Kennzahlen und Dimensionen entsprechend der Diskurswelt umgesetzt sind. Sollte der zu validierende Prototyp keine oder nur teilweise Zustimmung erfahren, findet ein Rückschritt bzw. eine Rückkopplung zu einer früheren Phase statt. Um die im Rahmen einer Rückkopplung anzusteuern Phase zu bestimmen, sind Verifikationen der Artefakte bereits durchschrittener Phasen notwendig. Abbildung 10 verdeutlicht das notwendige Vorgehen im Rahmen der Validierungsphase. Wird der realisierte Build hingegen angenommen, so sieht das Vorgehensmodell den Übergang zur Spezifikationsphase vor.

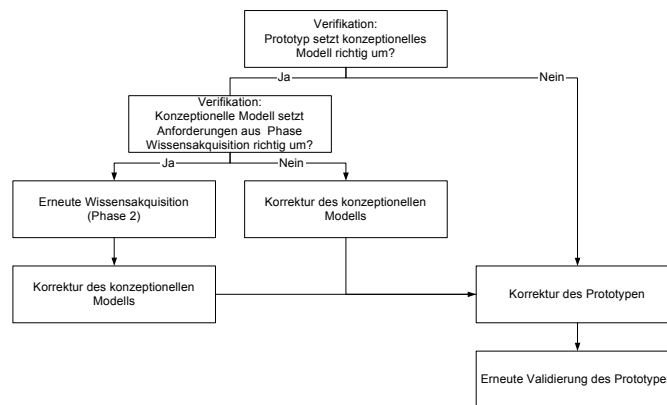


Abbildung 10: Vorgehen bei nicht akzeptiertem Prototyp

3.3 Spezifikation und Implementierung

Wird ein Prototyp nach der Validierung angenommen, so können die generierten logischen Strukturen als formalisierte Informationsnachfrage angesehen werden. Dem gegenüber steht ein Informationsangebot, welches überwiegend aus den operativen Systemen einer Unternehmung generiert und zusätzlich durch externe Informationen angereichert wird. Der Ausgleich zwischen Informationsbedarf und

Informationsangebot findet in Data-Warehouse-Systemen auf der Datenerfassungsebene in Form von ETL-Prozessen statt.

Wie bereits auf der Datenbereitstellungs- und Datenhaltungsebene besteht auch auf der Datenerfassungsebene die Möglichkeit zur konzeptionellen, logischen und physischen Modellierung [Vas⁺02a; TrLu03]. Analog zu den vereinfachten konzeptionellen Modellen, welche im Rahmen der Informationsbedarfsanalyse angewandt wurden, können konzeptionelle ETL-Modelle zur Kommunikation mit den Administratoren der operativen Systeme herangezogen werden.

Das Konzept zur konzeptionellen Modellierung von ETL-Prozessen von Vassiliadis et al. sieht vor, dass vor Beginn des Modellierungsprozesses eine Aufnahme der Benutzeranforderungen sowie eine strukturelle und inhaltliche Analyse der Datenquellen stattfindet [Vas⁺02a]. Die Benutzeranforderungen bezüglich des zu modellierenden ETL-Prozesses liegen in Form der Tabellendefinition des logischen Modells bereits vor (s.o.) und werden im Folgenden als Datenkonsumenten („Data Consumer“) bezeichnet. Dem gegenüber stehen die Datenquellen (interne und externe) als Datenlieferanten („Data Provider“).

Die Erstellung des konzeptionellen Modells des ETL-Prozesses erfolgt in einem dreistufigen Vorgehen [SiVa03]. In Schritt eins müssen adäquate Datenlieferanten gefunden und ausgewählt werden. Im Anschluss daran müssen die Liefer- und Mappingbeziehungen zwischen ggf. mehreren potenziellen Datenlieferanten und den Datenkonsumenten konkretisiert werden. Dieser zweite Schritt stellt aufgrund der Heterogenität der Quellsysteme einen kritischen Punkt im Modellierungsprozess dar und erfordert umfassende Diskussionen mit den Administratoren der operativen Datenbanken sowie umfangreiche Testläufe, um eine inhaltlich anforderungsgemäße Befüllung der Zieltabellen sicherzustellen. Nach Abbildung des Mappingvorgangs können durch Annotation des Prozesses Laufzeitbeschränkungen, wie z.B. Informationen über Ausführung, Überwachung und Logging des Prozesses sowie über Ausnahmen und Fehlerbehandlungen erfasst werden. Hierbei können auch zusätzliche Anforderungen, z.B. bezüglich der Aktualität der Daten, erfasst werden, welche es bei der physischen Implementierung zu beachten gilt (z.B. das Aktualisierungsintervall).

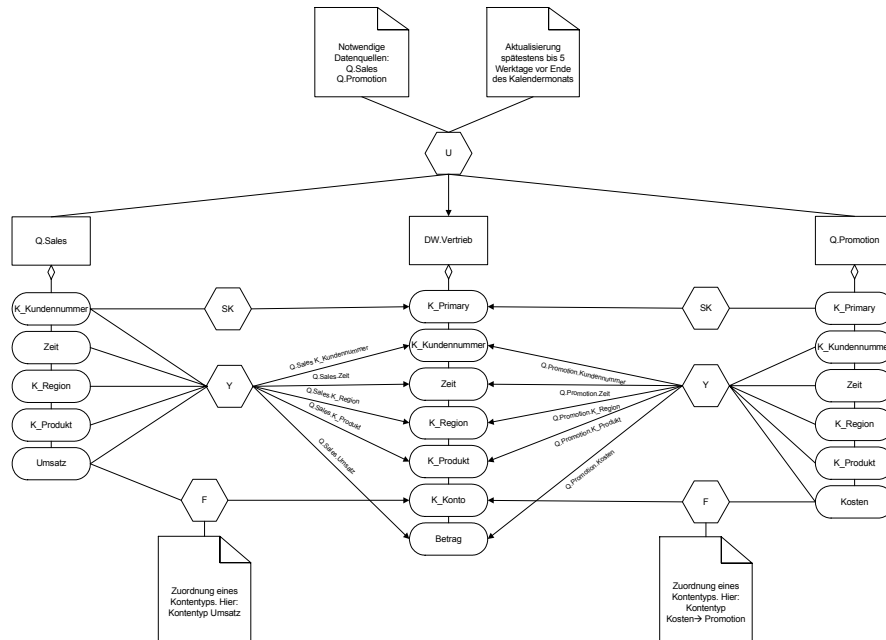


Abbildung 11: Beispielhafte Darstellung des konzeptionellen Modells eines ETL-Prozesses

Eine beispielhafte Darstellung des konzeptionellen Modells eines ETL-Prozesses kann Abbildung 11 entnommen werden (die grafische Notation wurde aus [Vas⁺02a] übernommen). Datenkonsument ist hierbei eine Faktentabelle (DW.Vertrieb) des Vertriebsinformationssystems, welche aus zwei verschiedenen Quellen (Q.Sales und Q.Promotion) versorgt wird. Im dargestellten Modell finden sich neben einfachen Mappingvorgängen (Y (Aggregationen)) auch weitere Transformationen. Einerseits wird ein einheitlicher Surrogatschlüssel vergeben (SK), um widersprüchliche Werte bei der Primärschlüsselvergabe der Faktentabelle zu vermeiden. Andererseits kommt eine Transformationsfunktion (F) zur Anwendung, welche einer Maßgröße Betrag in der Faktentabelle einen qualifizierenden Kontentyp zuordnet. So wird z.B. den Datensätzen aus der Quelldatenbank Promotion der Kontentyp Promotionkosten und den Datensätzen der Quelldatenbank Sales der Kontentyp Umsatz zugeordnet. Des Weiteren finden sich Annotationen für die Implementierung des Prozesses z.B. über die Ausführung oder die notwendigen Datenquellen (U).

Nach Vassiliadis et al. besteht ein ETL-Prozess auf logischer Ebene aus dem Datenfluss zwischen einer Datenquelle und dem Data-Warehouse sowie aus verschiedenen Aktivitäten, welche als logische Abstraktion physischen Codes angesehen werden können [Vas⁺02b; Vas⁺02c; Simi03]. Grundbestandteil des logischen Modells sind ETL-Aktivitäten, welche auf mehreren Ebenen betrachtet werden können. Aus Sicht des Metamodells bestehen Aktivitäten aus mehreren

elementaren Bestandteilen, wie z.B. dem Namen, Input- und Outputgrößen und dem Verhältnis zwischen diesen. Instanzen dieser Meta-Aktivitäten werden als Vorlage- oder Standard-Aktivitäten bezeichnet (z.B. Push, Join, Not Null Check usw.). Standard-Aktivitäten können weiter an die Anforderungen eines konkreten ETL-Prozesses angepasst werden. Diese angepassten Aktivitäten gelten dann als Instanz der Vorlagenebene.

Die Darstellung des logischen ETL-Prozesses erfolgt in so genannten ETL-Szenarios, welche aus einer Sequenz von Aktivitäten bestehen, die den Datenfluss zwischen den Quell- und Zieldatensätzen skizzieren. Abbildung 12 zeigt das ETL-Szenario des bereits vorgestellten Beispiels zum konzeptionellen ETL-Modell. Hierbei werden aus den Quelldatenbanken (Q.Sales, Q.Promotion) die in Frage kommenden Datensätze über FTP in Tabellen der Data-Staging-Area geladen (DS.Sales, DS.Promotion). Danach folgen mit der Zuordnung eines einheitlichen Surrogatschlüssels und dem Lookup der zutreffenden Kontenbezeichnung zwei ETL-Aktivitäten, wobei fehlerhaft verarbeitete Datensätze in Log-Dateien erfasst werden. Die abschließende Aktivität vereinigt die Datensätze aus den beiden Datenquellen und lädt diese in die Data-Warehouse-Tabelle (DW.Vertrieb).

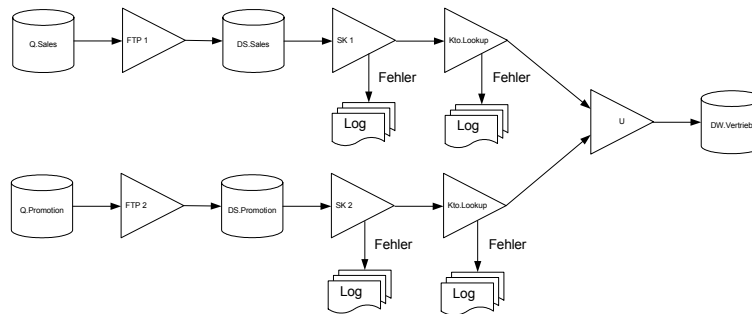


Abbildung 12: Darstellung eines ETL-Szenarios (logische Ebene)

Auf physischer Ebene werden logische ETL-Aktivitäten durch Programmcode konkretisiert, wobei auch dieser z.B. in Form von Struktogrammen, Programmablaufplänen, Pseudocode usw. [StHa05] konzeptionell abgebildet werden kann.

4 Fazit

In dem vorliegenden Aufsatz wurde eine Methode zur benutzerorientierten Entwicklung unternehmensweiter Data-Warehouse-Systeme entworfen. Der Schwerpunkt lag dabei auf einer Systemzerlegung, die die Aufteilung eines unternehmensweiten Data-Warehouse-Systems in inkrementell zu realisierende Builds vornimmt sowie einem Vorgehensmodell, welches Phasen und Entwicklungsergebnisse definiert. In dem von den Autoren durchgeführten Projekt EiSFach hat

sich die Methode als nutzbringend und handhabbar erwiesen [GoBu04a; GoBu04b; Goek04b]. Ob sich diese Nutzenpotentiale auch in einem komplexeren betrieblichen Umfeld realisieren lassen, sollte durch Anwendung der vorgeschlagenen Vorgehensweise und Methode geprüft werden.

Konkrete Techniken zur Erhebung der Benutzeranforderungen wurden nur am Rande betrachtet. Mögliche Techniken sollten im Rahmen einer Weiterentwicklung der Methode konkretisiert und den verschiedenen Entwicklungsphasen zugeordnet werden. Hierfür könnten Kontingenzmodelle zur situationsspezifischen Auswahl von Methoden und Techniken der Informationsbedarfsanalyse aufbereitet werden. Dies würde erlauben, eine sinnvoll begründete Auswahl aus der Vielzahl der zur Verfügung stehenden Techniken zu treffen. Zusätzlich sollten die Techniken zur Transformation der Entwicklungsergebnisse stärker formalisiert werden, woraus sich die Möglichkeit zur Werkzeugunterstützung ergeben könnte.

Darüber hinaus wurde mit dem Informationsbedarf zwar die wohl wichtigste Anforderung an Data-Warehouse-Systeme in den Mittelpunkt gestellt. Andere Anforderungen – wie z. B. Performance, Usability und Wartbarkeit – wurden nicht betrachtet. Dabei bietet der in Teil 2 dargestellte Bezugsrahmen Ansatzpunkte für die Modellierung und Implementierung auch dieser Anforderungen (bspw. lassen sich Performancegewinne alternativ auf der Ebene der logischen MDDS modellieren (Partitionierung und Fragmentierung der Tabellen) oder durch Techniken der Anfrageoptimierung (Indexstrukturen) auf physischer Ebene implementieren). Diese weitergehenden Anforderungen und zwischen ihnen auftretende Zielkonflikte sollten bei einer Weiterentwicklung der Methode im Rahmen eines umfassenden Anforderungsmanagements stärker Beachtung finden.

Literatur

- [Abe⁺00] Abello, A.; Samos, J.; Saltor, F.: A Data Warehouse Multidimensional Data Models Classification. Technical Report LSI-2000-6. Dept. Llenguages y Sistemas Informáticos (Universidad de Granada). Granada, 2000.
- [Bey⁺99] Beynon-Davies, P.; Tudhope, D.; Mackay, H.: Information systems prototyping in practice. In: *Journal of Information Technology* 14, 1999: S. 107–120.
- [Blas00] Blaschka, M.: FIESTA: A Framework for Schema Evolution in Multidimensional Databases. Dissertation, Technische Universität München, 2000.
- [BoU100] Böhnlein, M.; Ulbrich vom Ende, A.: Grundlagen des Data Warehousing: Modellierung und Architektur. *Bamberger Beiträge zur Wirtschaftsinformatik*, Nr. 55. Bamberg, 2000.
- [BrRo01] Browne, G.; Rogich, M.: An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques. In: *Journal of Management Information Systems* 17 (4), 2001: S. 223–249.

- [BuFo02] Bulos, D.; Forsman, S.: Getting started with ADAPT – OLAP Database Design. White Paper. Symetry Corp: San Rafael, 2002.
- [Bulo98] Bulos, D.: OLAP Database Design – A new Dimension. In: Chamoni, P.; Gluchowski, P. (Hrsg.): Analytische Informationssysteme – Data-Warehouse, OLAP, Data-Mining. Springer: Berlin et al., 1998, S. 249–262.
- [Eick01] Eicker, S.: Ein Überblick über die Umsetzung des Data-Warehouse-Konzepts aus technischer Sicht. In Schütte, R.; Rothowe, T.; Holten, R. (Hrsg.): Das Data Warehouse Managementhandbuch. Springer: Berlin et al., 2001, S. 65–79.
- [Flo⁺97] Floyd, C.; Krabbel, A.; Ratuski, S.; Wetzel, I.: Zur Evolution der evolutionären Systementwicklung: Erfahrungen aus einem Krankenhausprojekt. In: Informatik Spektrum 20 (1), 1997: S. 13–20.
- [Goek04a] Goeken, M.: Anforderungsmanagement bei der Entwicklung von Data-Warehouse-Systemen. Ein sichtenspezifischer Ansatz. DW2004 – Data Warehousing und EAI. Friedrichshafen, 2004.
- [Goek04b] Goeken, M.: Referenzmodellbasierte Einführung von Führungsinformationssystemen. In: Wirtschaftsinformatik 46 (5), 2004: S. 353–365.
- [GoBu04a] Goeken, M.; Burmester, L.: Entwurf und Umsetzung einer Business-Intelligence-Lösung für ein Fakultätscontrolling. In: Chamoni, P. et al. (Hrsg.): Tagungsband Multikonferenz Wirtschaftsinformatik 2004, Band 2. Essen, 2004, S. 137–152.
- [GoBu04b] Goeken, M.; Burmester, L.: Vorgehensmodell zur evolutionären und benutzerorientierten Data-Warehouse-Entwicklung. In: Bauer, A. et al. (Hrsg.): Internationales Symposium: Data-Warehouse-Systeme und Knowledge-Discovery. Darmstadt, 2004, S. 53–64.
- [GoLi93] Goguen, J. A.; Linde, C.: Techniques for Requirements Elicitation. Proceedings of IEEE International Symposium on Requirements Engineering 1993. San Diego, 1993, S. 152–164.
- [Gol⁺98] Golfarelli, M.; Maio, D.; Rizzi, S.: Conceptual Design of Data Warehouses from E/R Schemes. Proceedings of the Hawaii International Conference On Systems Science. Kona, 1998.
- [GoRi98] Golfarelli, M.; Rizzi, S.: A Methodological Framework for Data Warehouse Design. Proceedings of the ACM first international Workshop on Data Warehousing and OLAP (DOLAP). Washington, 1998.
- [Hack98] Hackney, D.: Architectures and Approaches for Successful Data Warehouses. <http://datawarehouse.ittoolbox.com/documents/document.asp?i=815>, 1998, Abruf am 2004-09-12.
- [Hahn98] Hahne, M.: Logische Datenmodellierung für das Data Warehouse – Bestandteile und Varianten des Star Schemas. In: Chamoni, P.; Gluchowski, P. (Hrsg.): Analytische Informationssysteme – Data Warehouse, OLAP, Data Mining. Springer: Berlin et al., 1998, S. 103–122.

- [Hahn02] Hahne, M.: Transformation mehrdimensionaler Modelle. In: von Maur, E.; Winter, J. (Hrsg.): Vom Data Warehouse zum Corporate Knowledge Center. Physica-Verlag: Heidelberg, 2002, S. 399–420.
- [Hec⁺03] Heck-Weinhart, G.; Mutterer, G.; Herrmann, C.; Rupprecht, J.: Entwicklung eines Vorgehensmodells für DWH-Projekte bei der W&W AG. In: von Maur, E.; Winter, R. (Hrsg.): Data Warehouse Management. Das St. Galler Konzept zur ganzheitlichen Gestaltung der Informationslogistik. Springer: Berlin et al., 2003, S. 197–220.
- [Herd01] Herden, O.: Eine Entwurfsmethodik für Data Warehouses. Dissertation, Universität Oldenburg, 2001.
- [Hess97] Hesse, W.: Wie evolutionär sind die objektorientierten Analysemethoden? Ein kritischer Vergleich. In: Informatik Spektrum 20, 1997: S. 21–28.
- [Holt99] Holten, R.: Entwicklung von Führungsinformationssystemen. Ein methodenorientierter Ansatz. Dissertation, Universität Münster, 1999.
- [Jar⁺99] Jarke, M.; Jeusfeld, M.A.; Quix, C.; Vassiliadis, P.: Architecture and Quality for Data Warehouses: An Extended Repository Approach. In: Information Systems 24 (3), 1999: S. 229–253.
- [KiRo02] Kimball, R.; Ross, M.: The Data Warehouse Toolkit. John Wiley: New York, 2002.
- [Lehn03] Lehner, W.: Datenbanktechnologie für Data-Warehouse-Systeme: Konzepte und Methoden. dpunkt-Verlag: Heidelberg, 2003.
- [NoCr99] Nordbotten, J. C.; Crosby, M. E.: The effect of graphic style on data model interpretation. In: Information Systems Journal 9, 1999: S. 139–155.
- [Ong99] Ong, H.: The Evolution Of A Data Warehouse Architecture - One Size Fits All? WA Oracle User Group Conference 1999, Perth.
<http://www.aurora-consult.com.au/white.html>, 1999, Abruf am 2004-10-12.
- [Ortn95] Ortner, E.: Elemente einer methodenneutralen Konstruktionsprache für Informationssysteme. In: Informatik - Forschung und Entwicklung. 10 (3), 1995: S. 148–160.
- [Pera03] Peralta, V.: Data Warehouse Logical Design from Multidimensional Databases. <http://www.fing.edu.uy/inco/grupos/csi/esp/Publicaciones/2003/clei2003-vp.pdf>, 2003, Abruf am 2004-10-12.
- [PeRu03] Peralta, V.; Ruggia, R.: Using Design Guidelines to improve Data Warehouse logical Design. Proceedings of the International Workshop on Design and Management of Data Warehouses colocated with VLDB'03, Berlin.
http://www.fing.edu.uy/inco/grupos/csi/esp/Cursos/cursos_act/2003/DAP_SistDW/Material/DesGuidelines-VPRR.pdf, 2003, Abruf am 2004-10-12.
- [QuDe99] Quigley, E. J.; Debons, A.: Interrogative Theory of Information and Knowledge. Proceedings of the 1999 ACM SIGCPR conference on Computer personnel research. San Diego, 1999, S. 4–10.
- [Quix03] Quix, C. J.: Metadatenverwaltung zur qualitätsorientierten Informationslogistik in Data-Warehouse-Systemen. Dissertation, Technische Hochschule Aachen, 2003.

- [Sche98] Schelp, J.: Konzeptionelle Modellierung mehrdimensionaler Datenstrukturen. In: Chamoni, P.; Gluchowski, P. (Hrsg.): Analytische Informationssysteme – Data Warehouse, OLAP, Data Mining. Springer: Berlin et al., 1998, S. 263–275.
- [Sche00] Schelp, J.: Modellierung multidimensionaler Datenstrukturen analyseorientierter Informationssysteme. Dissertation, Universität Bochum, 2000.
- [Schi01] Schirp, G.: Anforderungsanalyse im Data-Warehouse-Projekt: Ein Erfahrungsbericht aus der Praxis. In: HMD 222, 2001: S. 81–87.
- [Simi03] Simitis, A.: Modeling and managing ETL Processes. Proceedings of the VLDB 2003 PHD Workshop. Berlin, 2003.
- [SiVa03] Simitis, A.; Vassiliadis, P.: A Methodology for the Conceptual Modeling of ETL Processes. Decision Systems Engineering Workshop (DSE'03), in conjunction with the 15th Conference on Advanced Information Systems Engineering (CAiSE '03). Klagenfurt / Velden, 2003.
- [StHa05] Stahlknecht, P.; Hasenkamp, U.: Einführung in die Wirtschaftsinformatik, 11. Auflage. Springer: Berlin et al., 2005.
- [TrLu03] Trujillo, J.; Lorán-Mora, S.: A UML Based Approach for Modeling ETL Processes in Data Warehouses. In: Song, I-Y. (Hrsg.): Proceedings ER 2003. Springer: Berlin et al., 2003, S. 307–320.
- [VaFr85] Valusek, J.R.; Fryback, D.G.: Information requirements determination obstacles within, among and between participants. Proceedings of the twenty-first annual conference on computer personnel research. Minneapolis, 1985, S. 103–111.
- [Vas⁺00] Vassiliou, Y. et al.: Data Warehouse Research: Issues and Projects. In: Jarke, M. et al. (Hrsg.): Fundamentals of Data Warehouses. Springer: Berlin et al., 2000, S. 15–21.
- [Vas⁺02a] Vassiliadis, P.; Simitis, A.; Skiadopoulos, S.: Conceptual Modeling for ETL Processes. Proceedings of the 5th International Workshop on Data Warehousing and OLAP (DOLAP 2002). McLean, 2002, S. 14–21.
- [Vas⁺02b] Vassiliadis, P.; Simitis, A.; Skiadopoulos, S.: On the logical Modeling of ETL Processes. Proceedings of the 14th conference on advanced information Systems Engineering (CaiSE'02). Toronto, 2002, S. 782–786.
- [Vas⁺02c] Vassiliadis, P.; Simitis, A.; Skiadopoulos, S.: Modeling ETL Activities as Graph. Proceedings of the 4th international Workshop on Design and Management of Data Warehouses (DMDW'2002). Toronto, 2002, S. 52–61.
- [WaWe02] Wand, Y.; Weber, R.: Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. In: Information Systems Research 13 (4), 2002: S. 363–376.